

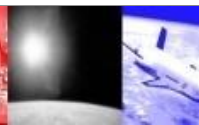
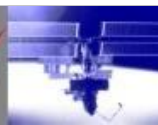
HOSC POIWG Splinter Session

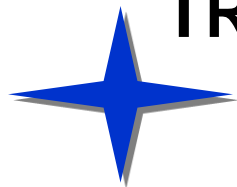
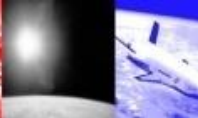
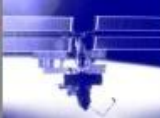
Gina Wade



Telecon: 1-844-467-6272

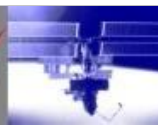
Passcode: 972352





TReK (Telescience Resource Kit)

Michelle Schneider
Jeff Lippincott



Telescience Resource Kit (TReK) Information for POIWG Splinter



January 28, 2015

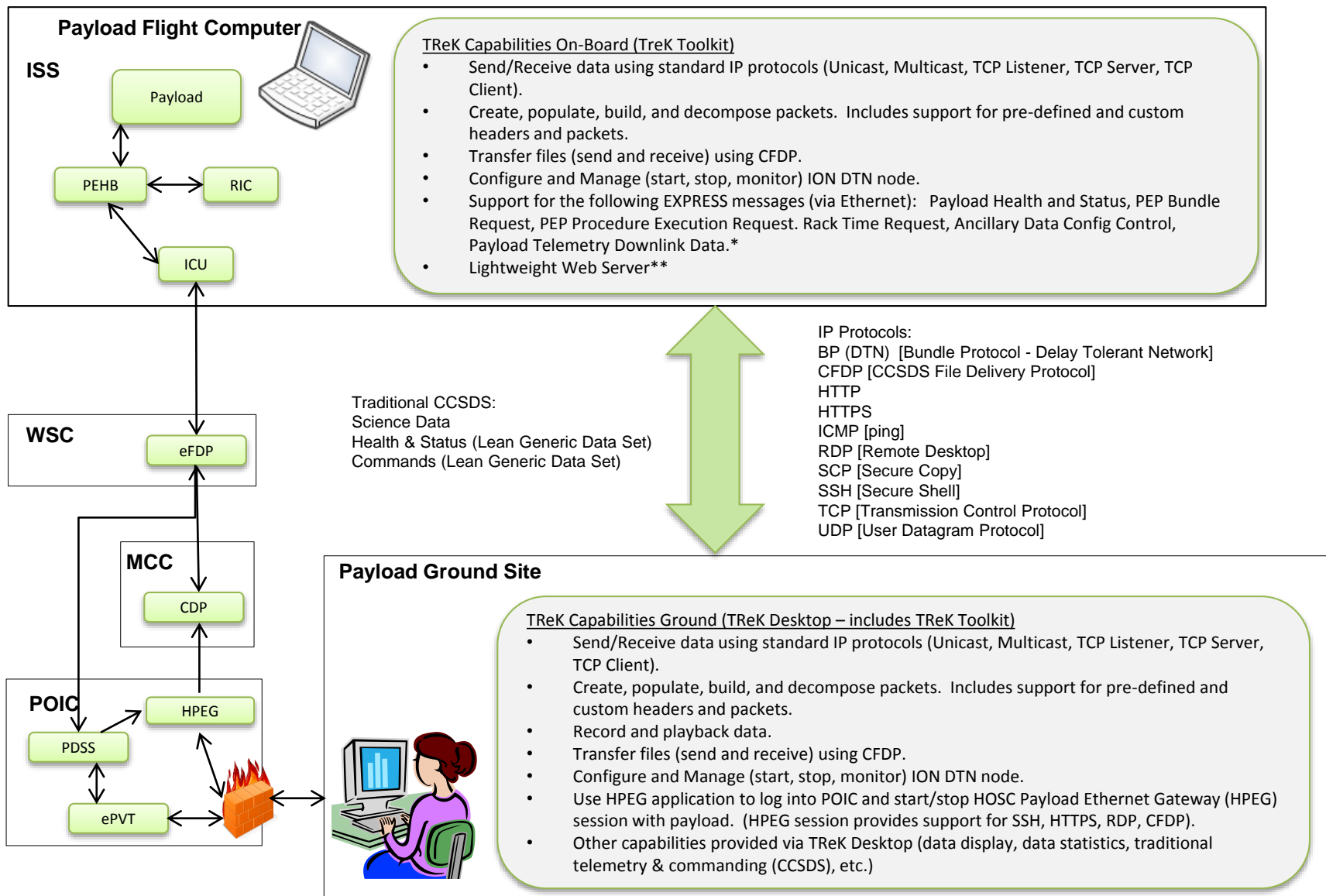


TReK Information

- TReK 4.2.0 onboard software verification has been completed.
 - Check with your Payload Integration Manager for information about leveraging off the TReK Verification.
- The TReK Demonstration Payload is scheduled to fly in early 2016 (Ready Date: January 27, 2016).
 - Payload is currently listed as reserve so we don't have an exact date the payload will be onboard.
 - Demonstrate the TReK software running on-orbit on an ISS provided T61p laptop.
 - Pathfinder Payload for NAS and DTN.
 - Current Uplink Plan
 - Uplink the TReK Software to the Payloads Network Address Storage (NAS).
 - Crew initiates image install from Payload NAS using PXE-Boot. (Crew time for the T61p Laptop set up and software transfer from NAS is currently reserve time).
 - Run performance tests for the IP protocols described in SSP 52050.
 - Provide the capability to support live demonstrations during ISS Payload Conferences.
 - Provide the TReK Demonstration Payload as a training tool in the Huntsville Operations Support Center (HOSC) On Demand Test Environment for payloads seeking to experience operating with IP protocols.



TReK Information





TReK Information

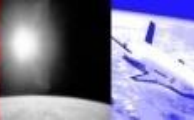
- TReK 4.3.0 will be delivered April 29, 2016. This release will contain the following:
 - Updates to TReK HPEG software to align with the latest HOSC PGUIDD updates for Ku IP.
 - A new TReK HPEG API to provide programmatic access to TReK HPEG information.
 - Updates to the TReK CFDP software to support filestore directives when running native CFDP.



Point of Contact

Name: Michelle Schneider
Phone: (256) 544-1535
Fax: (256) 544-9353
E-Mail: Michelle.Schneider@nasa.gov
Address: Building 4663, EO50
Huntsville, AL 35812

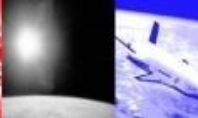
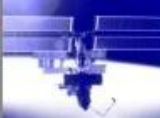
TReK E-Mail: trek.help@nasa.gov
TReK Web Site: <http://trek.msfc.nasa.gov>



AEON Portal Demonstration

George Ritter
Gina Wade



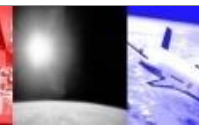
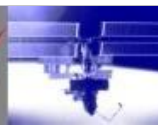


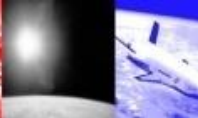
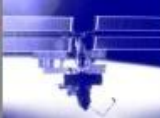
ShareKnowledge Demonstration

April Hargrave

Jessica McCauley

Gina Wade

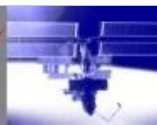




ODTE (On-Demand Test Environment) Presentation



Robin Strong

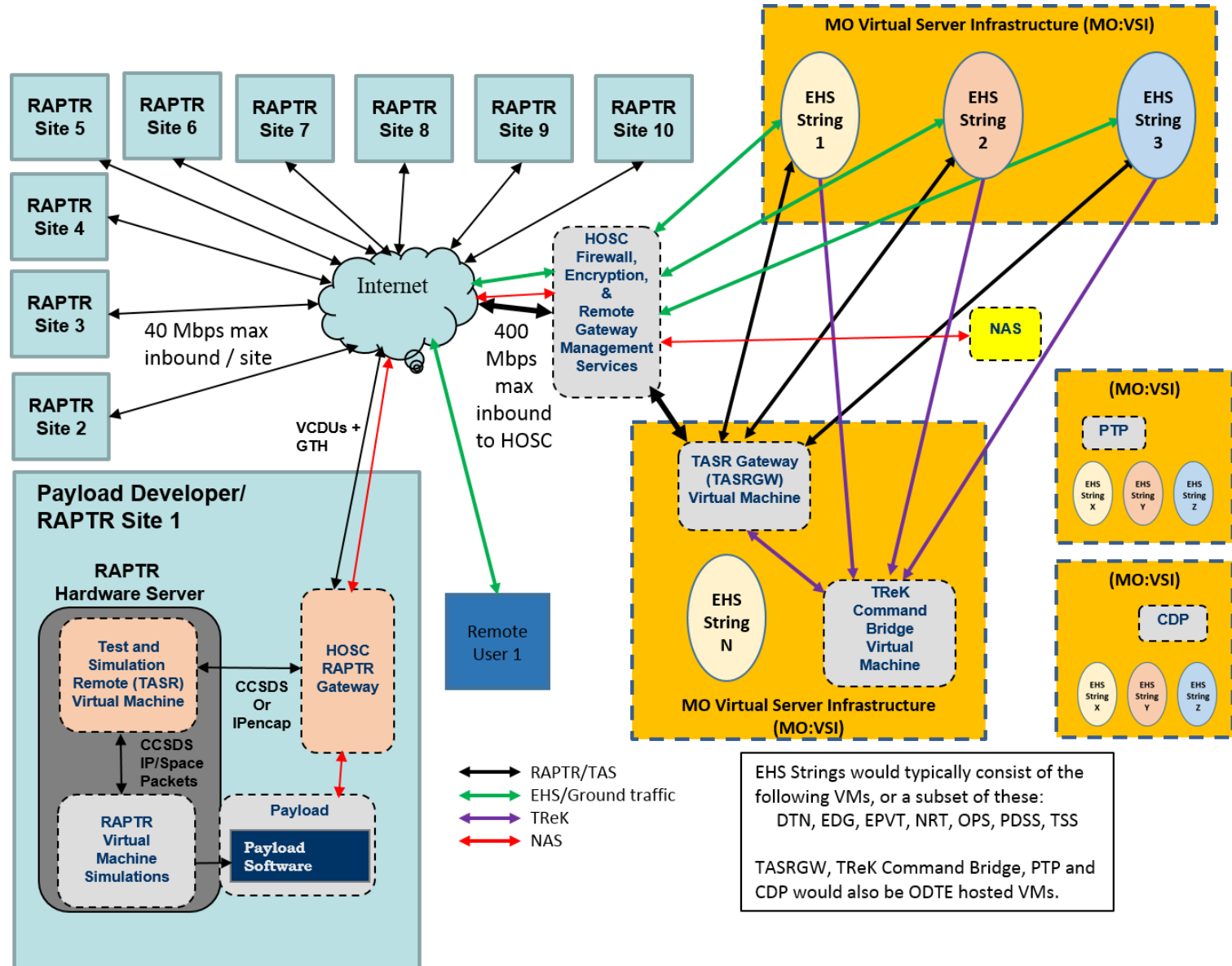


On-Demand Test Environment (ODTE) for Payload Developers

HOSC POIWG Splinter Session

Robin Strong

- Historically, new payloads have experienced challenges the first time they attempt to perform any kind of testing with the HOSC.
 - High fidelity testing with the HOSC IVV environment can only occur during the prime shift, and only one payload at a time can use this equipment string.
 - ODTE provides a lower fidelity testing capability which mitigates risk by exercising payload and ground support interfaces early in the payload's lifecycle.
- Cost savings are realized when Payload Developers can perform ISS integration testing and HOSC integration testing while the payload is at the Payload Developer's site.
 - Schedule and shipping risks are also avoided.
- Flight readiness preparation is facilitated by simulating end-to-end flight-like interfaces and message traffic between the payload and its ground test hardware.
- The HOSC's goal is to support concurrent testing from at least 10 payloads at any time of day, with minimal assistance from Operations Center staff.



- Boeing's Remote Advanced Payload Test Rig (RAPTR) is composed of two modules, the Command and Data Handling (C&DH) Module and the Analog & Discrete (A&D) Module, plus associated cabling.
 - RAPTR C&DH Module provides the MIL-STD-1553, Ethernet, and High Rate Data Link (HRDL) interfaces for the USOS, EXPRESS Rack, and ELC systems.
 - RAPTR A&D Module provides the analog and discrete interfaces for the EXPRESS Rack, ELC, JEM EF, and Columbus Back Porch system.

A&D Module



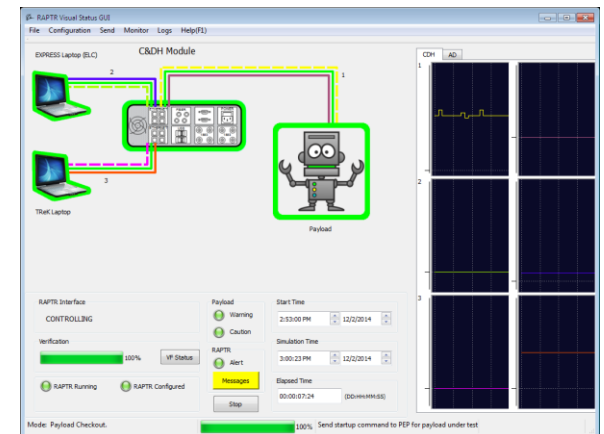
C&DH Module

RAPTR System

A&D Module
(Back)



CDH Embedded System
(Back)

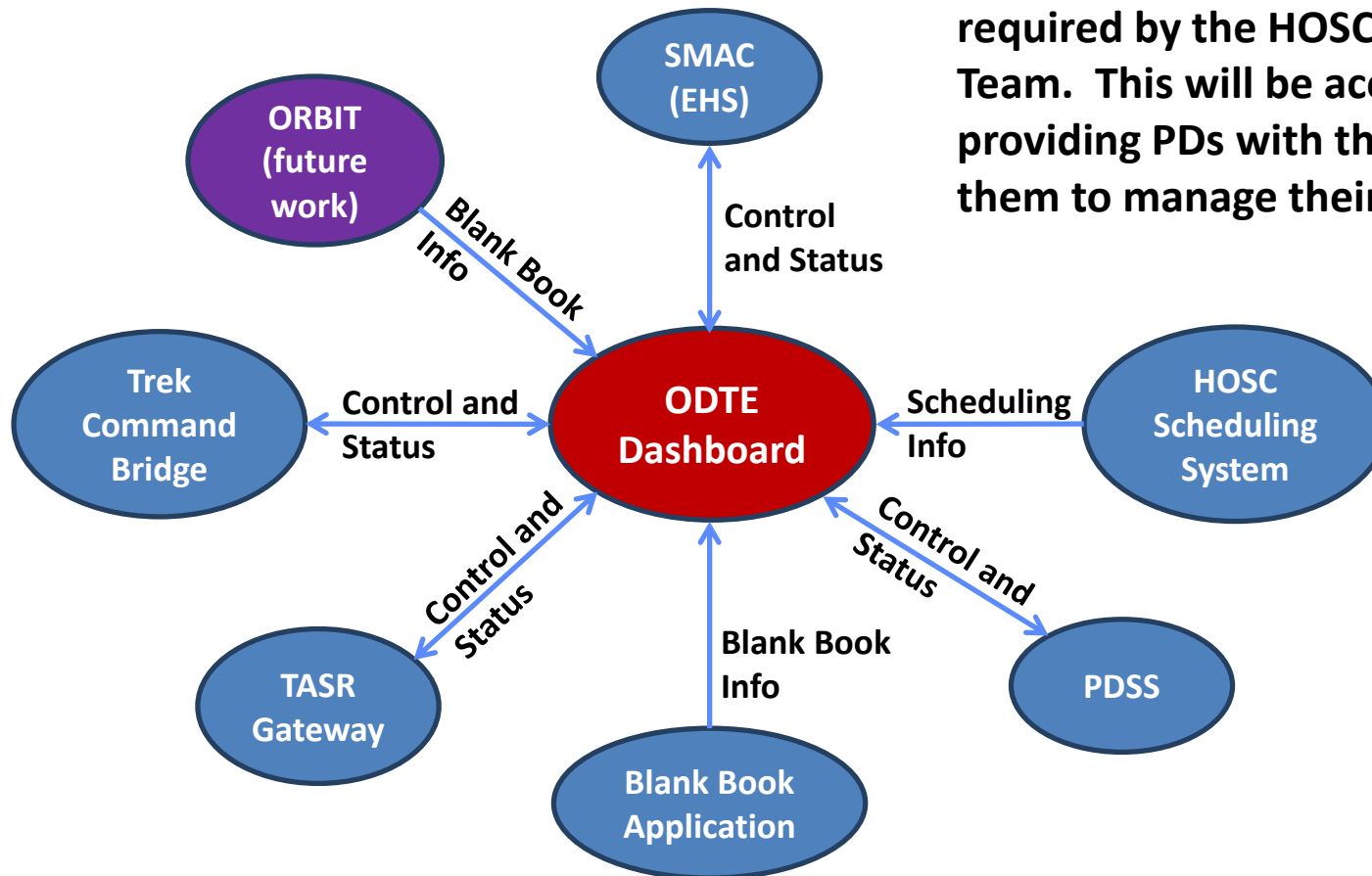


RAPTR User Display

1. (FY16) Payload developer is an established user; all payload commanding and telemetry is already defined in HOSC databases; a RAPTR and RAPTR Gateway are operational at the developer's site.
 - All information required by POIC and the ODTE Dashboard is present to instantiate the ODTE string.
2. (FY16) Payload developer plans to use IP encapsulation exclusively, AND does not need command or telemetry databases. (Example: Remote access to payload laptop.)
 - Developer connects to HOSC using standard HOSC VPN services rather than a RAPTR Gateway.
 - A generic ODTE environment is instantiated which supports IPEncap and DTN.
3. (FY17) Payload developer has completed their Ground Data Services (GDS) Blank Book, but does not have commanding or telemetry defined in PDL or in HOSC databases; a RAPTR and RAPTR Gateway are operational at the developer's site.
 - ODTE Dashboard or AEON Portal provides an interface to define message formats.
 - After message formats have been defined locally, databases and local tables can be generated, then ODTE virtual machines can be instantiated.
 - The functionality provided by this use case and use case 2 will support IP Encapsulated Health and Safety packets.
4. (FY17) Payload developer needs to perform CoFR testing after the transition to a new build or new database has occurred.
5. (Future Opportunity) Payload developer does not yet have GDS Blank Book and does not have a RAPTR. They request access to ODTE for familiarization and training using the TReK Demo Payload.
 - Developer connects to HOSC using standard HOSC VPN services rather than a RAPTR Gateway.
 - A generic ODTE training environment is instantiated.

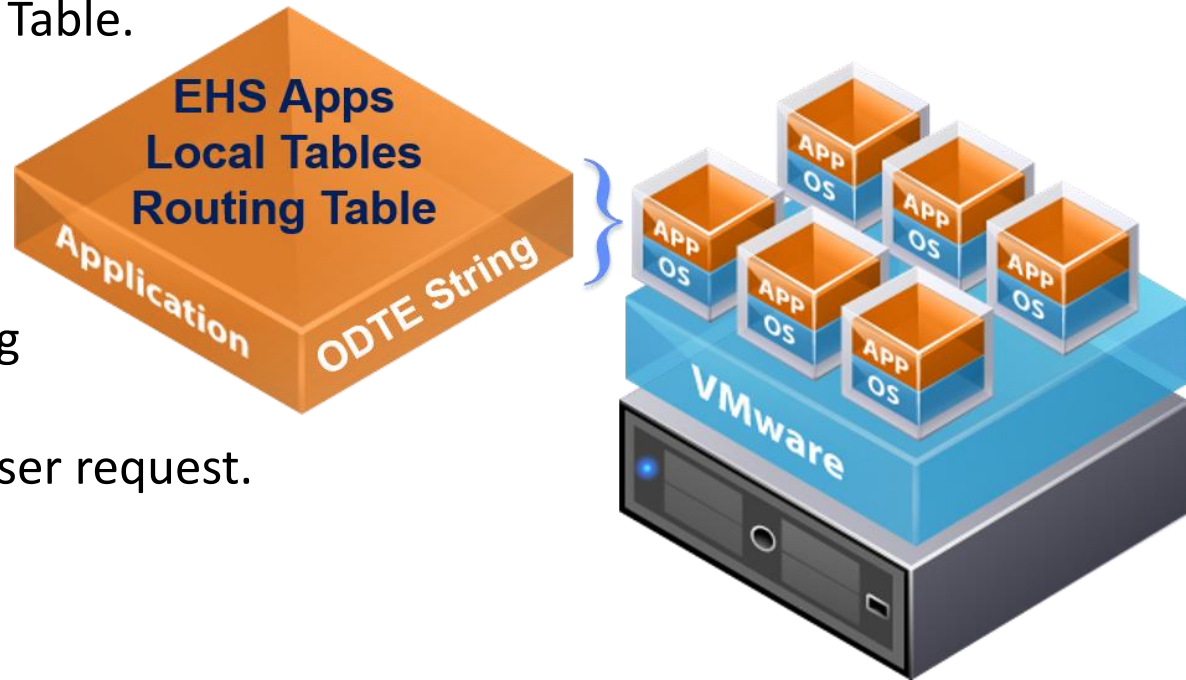
The ODTE Dashboard will be the primary interface for Payload Developers (PDs) to plan and manage the HOSC resources required to support remote payload testing.

A major goal of the ODTE Dashboard is to minimize the amount of interaction required by the HOSC Integrated Support Team. This will be accomplished by providing PDs with the tools that will allow them to manage their private EHS string.



ODTE Dashboard Tasks

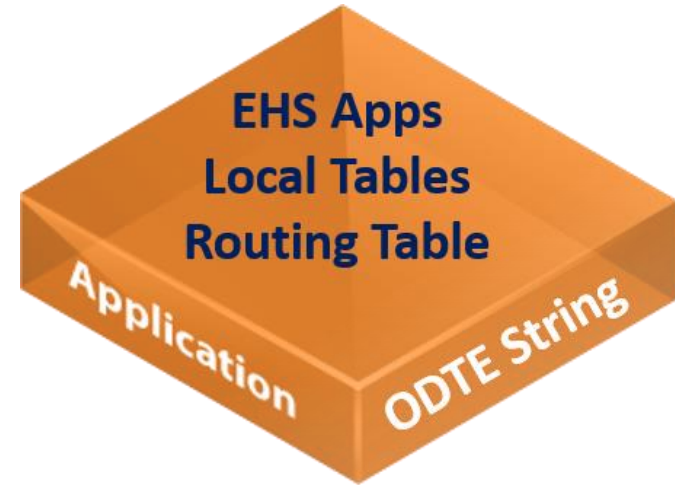
1. Parse the GDS Blank Book content.
2. Instantiate EHS Apps and Local Tables on ODTE virtual machines.
3. Initiates database instantiation in the IV&V RAC at user request.
4. Configure the EHS Routing Table.
5. Start/Stop EHS services at user request.
6. Provide status of ODTE components.
7. Enable/Disable NRT logging at user request.
8. Reset packet counters at user request.



NOTE: Instantiation of an ODTE string occurs once when it is added to the Virtual Server Infrastructure. The string is stored in the VMFS on the SAN datastore and is reloaded as needed. The string is not rebuilt each time the developer logs into the ODTE Dashboard.

Payload Developer Tasks

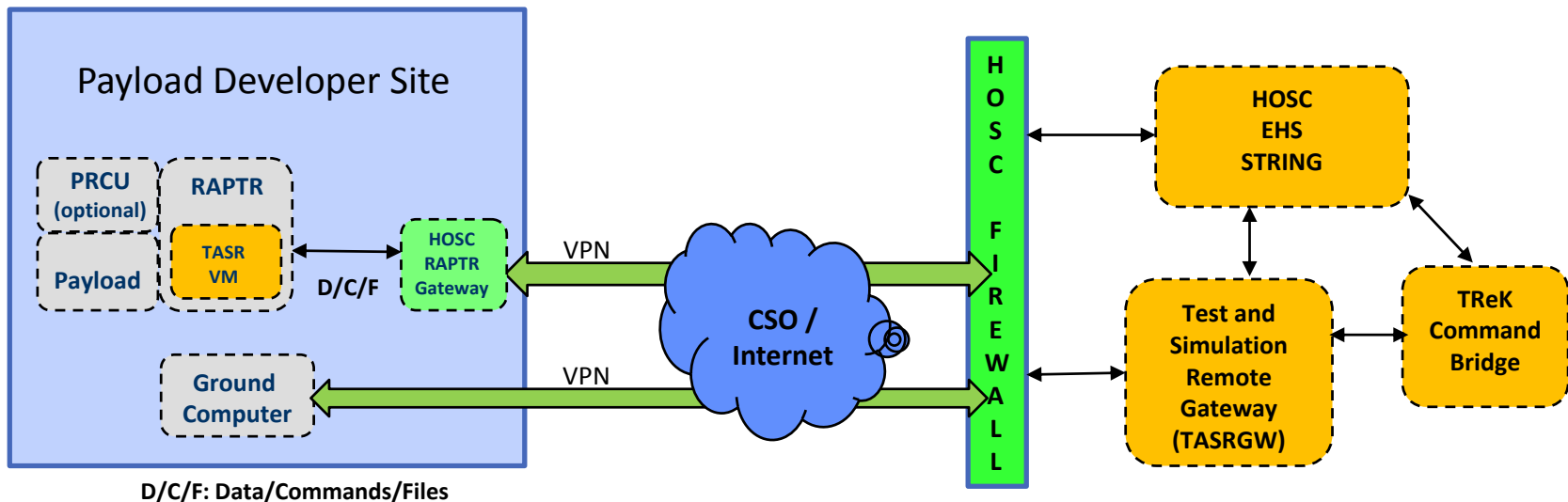
1. Populate ORBIT website and Ground Data Services (GDS) Blank Book.
2. Start/stop ODTE dashboard session.
3. Select payload if they have access to more than one payload.
4. Select role if they have access to more than one role.
5. Specify metadata for each test activity. (optional)
6. Specify desired version of full increment database **<OR> define and populate a private test database.**
7. Enabled or disable NRT Logging.
8. Check a box on the Dashboard to indicate that they are preparing for CoFR. This would activate NRT logging and any other features we believe would be appropriate during this phase of testing.



Network Engineering Team Tasks

Manage RAPTR Gateway and HOSC Firewall Configurations

1. Configure Check Point 2200 devices deployed as RAPTR Gateways.
2. Configure HOSC firewall settings for VPN tunnel between payload developer site and the HOSC.
3. Ship RAPTR Gateways to payload developers.

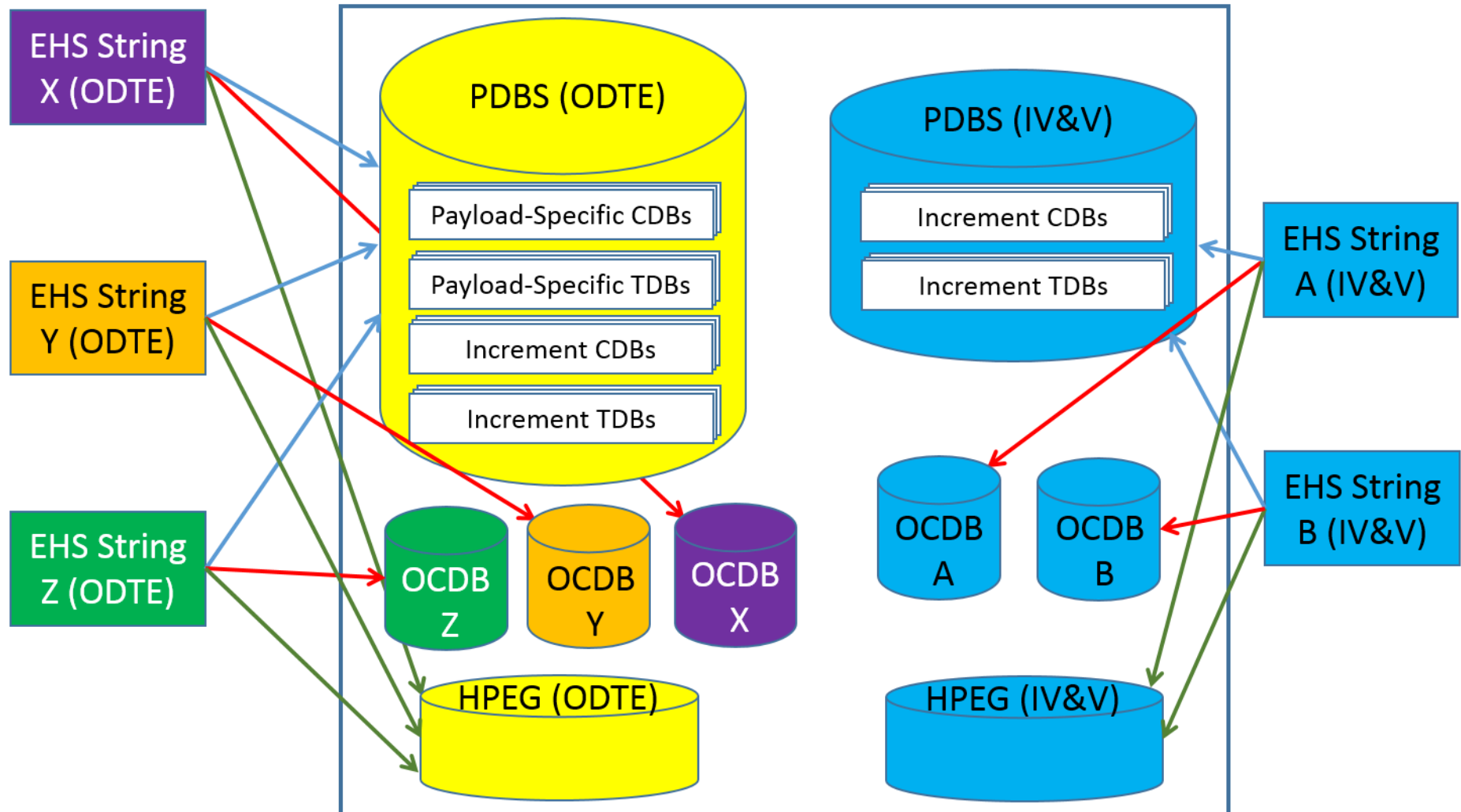


Network Attached Storage (NAS) functionality which is nearly identical to that provided on the ISS will be accessible by ODTE users, but we will not be supporting PXE Boot through the HOSC firewall since the packet latency and network routing would be significantly different from what is experienced on the ISS.

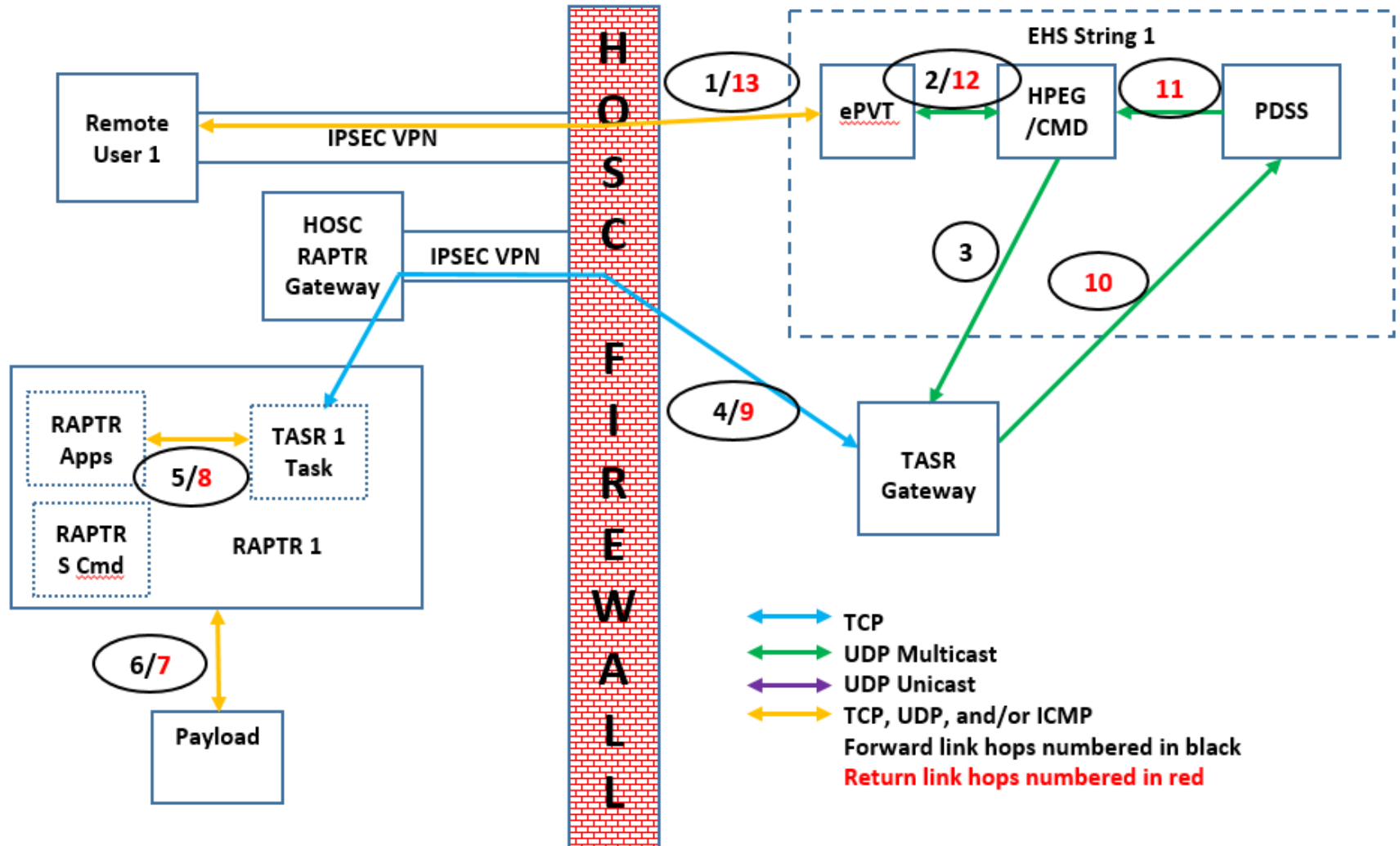
The ODTE will support CFDP file transfers, but will not emulate or support PLMDM file uplink via legacy Drop Box capabilities.

ODTE Database Architecture

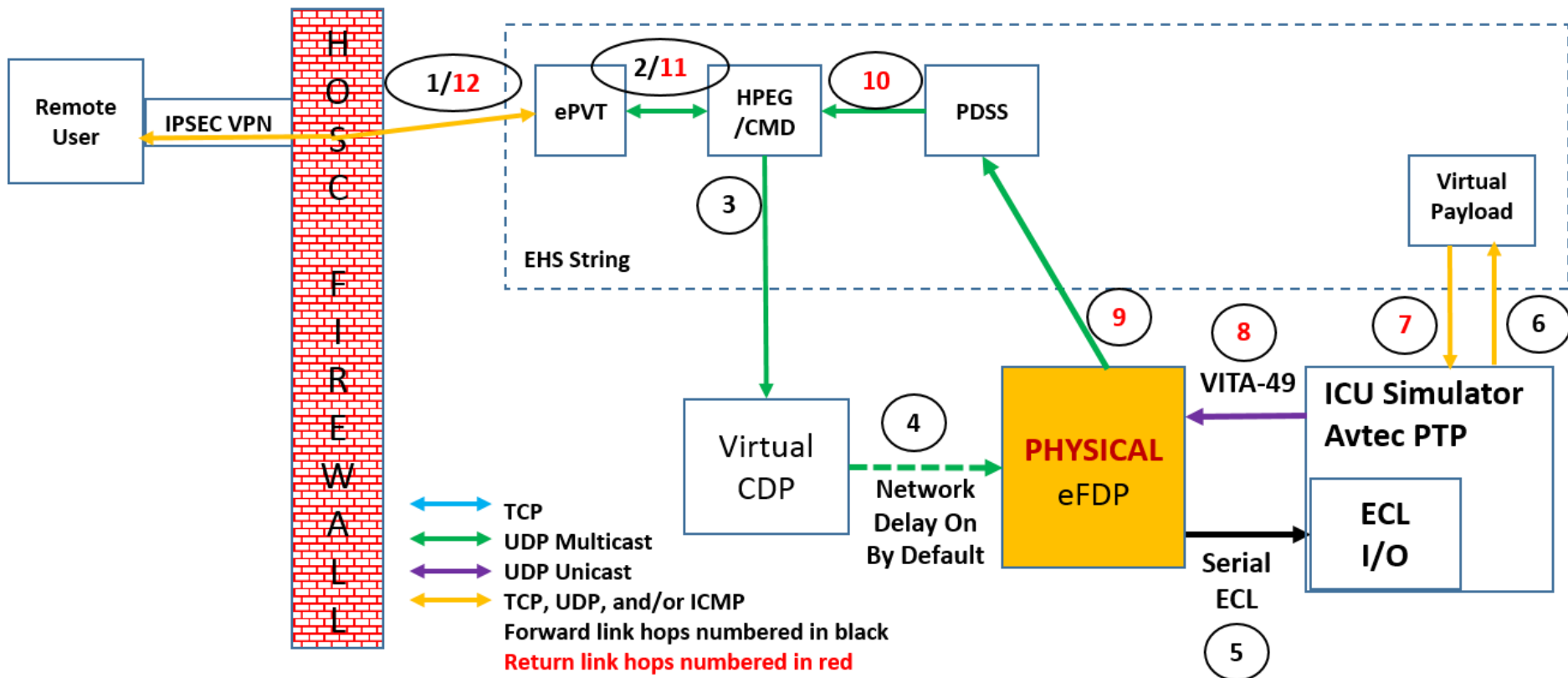
IV&V Oracle RAC



Low Fidelity Testing Without ECL Conversion

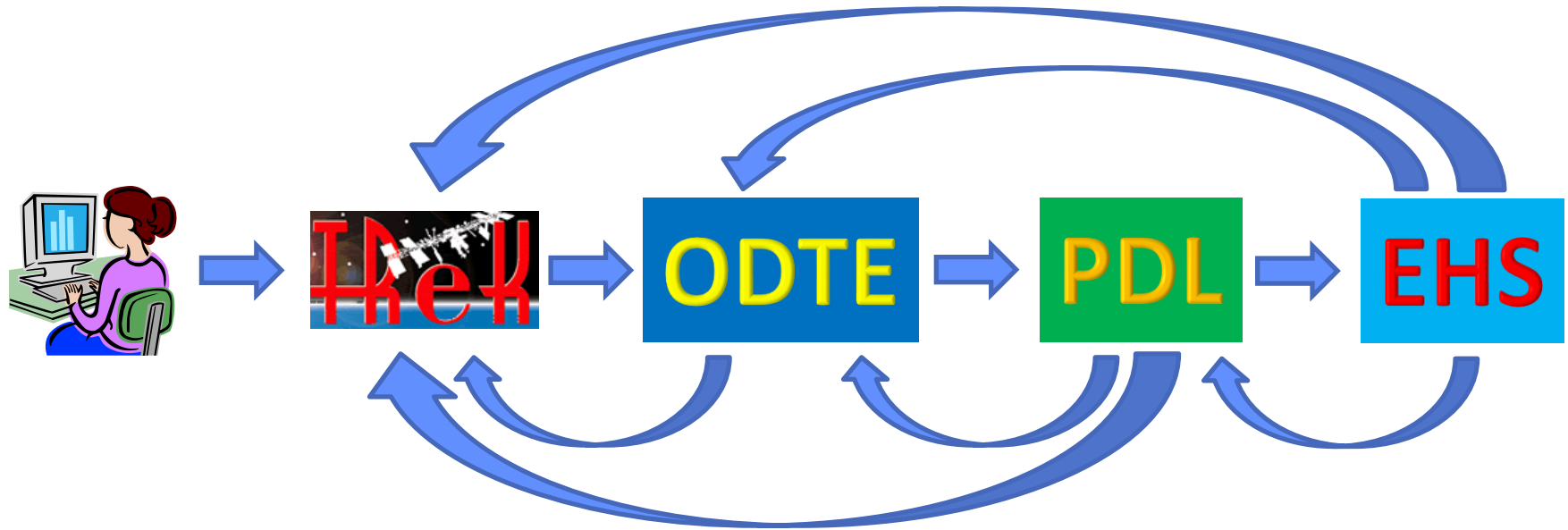


Higher Fidelity Testing With ECL Conversion

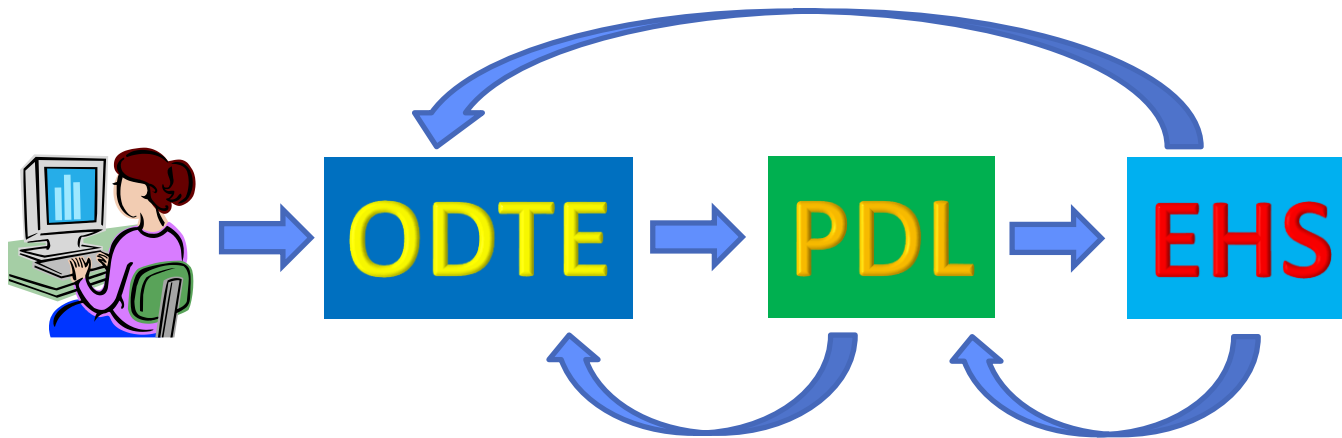


Viable for One Virtual Payload at a Time

- The TASR Gateway will accept TASR input from multiple payload developer sites, parse the headers, then disseminate to the appropriate virtualized PDSS server using the current IIGOR multicast format.
 - A separate virtual PDSS server will be allocated for each remote RAPTR gateway (e.g., RAPTR3 ⇔ PDSS3).
- As in mission configuration, the virtual PDSS will extract the S-Band and Ku IP Services content, attach an EHS header, then forward the packets to a virtual EHS string for further processing.
- Wherever possible, customized behavior for components in the ODTE EHS string will be implemented through configuration files, without requiring any customization of the individual software applications.
- Software development will be needed for the ODTE Dashboard (a web interface for the users and system managers), virtualization of the PDSS, a new version of the TReK Command Bridge, and updates to the TASR Gateway to interface with the ODTE Dashboard.

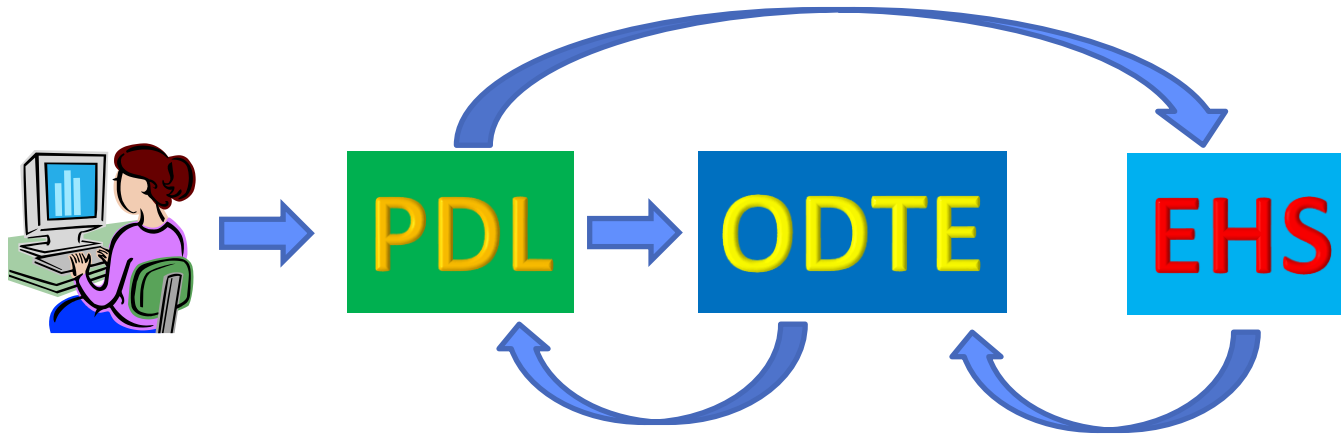


Use Case 1: The typical payload development team starts out using TReK exclusively. When they are allocated a RAPTR, they connect to ODTE and import their DB content from TReK. When they have finalized their message formats in ODTE, they export that DB content and submit it to PDL. PDL generates the formal delivery to EHS for flight. A database correction can be applied at any stage and migrated to any other database.



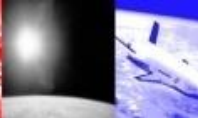
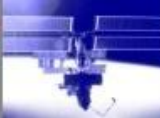
Use Case 2: The payload development team starts out using ODTE when they are allocated a RAPTR, and define their DB content using spreadsheets or a Data Wizard GUI. When they have finalized their message formats in ODTE, they export that DB content and submit it to PDL. PDL generates the formal delivery to EHS for flight.

A database correction can be applied at any stage and migrated to any other database.



Use Case 3: The payload development team already has their legacy message formats in PDL. When they are allocated a RAPTR for new work, they populate their ODTE database from a payload-specific PDL export. They update their DB content on ODTE using spreadsheets or a Data Wizard GUI. When they have finalized their message formats in ODTE, they export that DB content and submit it to PDL. PDL generates the formal delivery to EHS for flight. A database correction can be applied at any stage and migrated to any other database.

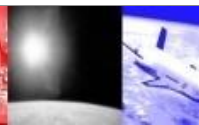
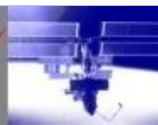
- FY16 Q2 – Q3: support for Use Cases 1 and 2
 - Design Review – March 21, 2016
 - Core functionality of ODTE Dashboard
 - PDSS virtualization
 - Updates to TReK Command Bridge and TASR Gateway
 - Instantiate 3 virtual EHS strings
 - Instantiate database support in IV&V RAC
 - Update DNS and firewall configuration
- FY16 Q4 - FY17Q1: support for Use Case 3
 - Design Review – August 2016
 - IV&V of Use Case 1
 - Instantiate remaining virtual EHS strings
- FY17 Q2 – Q3:
 - Final release of software under this ECR
 - IV&V of Use Case 2 in Q2
 - IV&V of final release in Q3



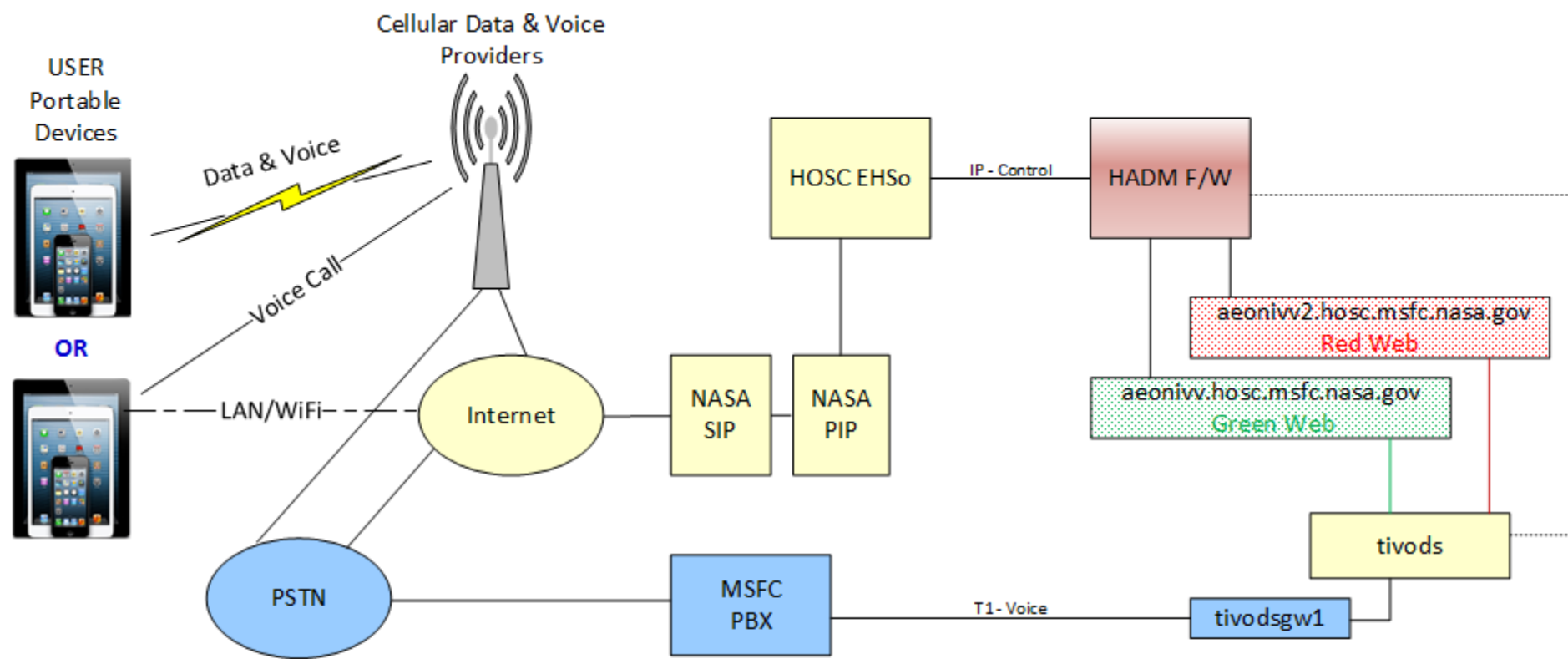
Mobile IVoDS Presentation and Demonstration



Steve Calvelage



Mobile VoDS Functional Diagram



Types of connections

Voice / Telephone:

- Cellular Call
- Hardline / Desk Phone
- Soft Phone Application

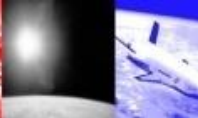
Control / Web Client

- Web Browser
- Cellular Data
- Wifi
- Hardline / LAN



PBX Phone numbers (24)

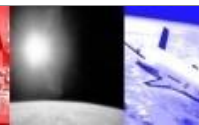
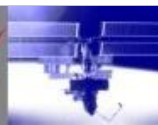
256 544 9725
256 544 9726
through
256 544 9748



Ku IP Lessons Learned Presentations

Sparky Goodman

Larry Schneider



NASA HOSC Ku-IP Services (Ku-IPS)

Characterization & Recommendations

*** POIWG v0.3 ***

HOSC Ku-IPS Lessons Learned (1)

- Since users do ETE development testing using the HOSC JSC testbed, the HOSC becomes a defacto collector of lessons learned for payload Ku-IP service usage.
- As this information emerges, the HOSC is providing a Lessons Learned for collecting and sharing Ku-IPS characterization and recommendations.
- The HOSC plans to characterize [Ku-IPS Baselined Protocols](#) defined in ECR HM-3430 and HM-3464 as available to all PDs.
 - PING, SSH (22/TCP), HTTPS (443/TCP), RDP (3389/TCP), LTP (1113/UDP), BP (4556/UDP) and CFDP (4560/UDP).
 - TCP must originate from the ground.
 - TCP originating on-board will be available as of Increment 50.
- PDs have the lead to characterize their unique data flows.
 - With the PD's permission, the HOSC will incorporate PD data flow characterization and recommendations in the Ku-IPS Lessons Learned.

HOSC Ku-IPS Lessons Learned (2)

- The HOSC is establishing a Ku-IPS mailing list.
- Ku-IPS Lessons Learned updates will be distributed twice per year after the POIWG.
 - Interim updates distributed when necessary.
 - <https://aeonsp.hosc.msfc.nasa.gov/sites/Remotes/KUIPLL/SitePages/Home.aspx>
- Since this information is most valuable during the design phase, forward work is to provide relevant information to the OZ front doors and to new payload Kickoff meetings for those interested in using Ku-IP Services. This topic will be worked as part of OZ/POI RISE improvements.

iPerf3

<https://iperf.fr/>

- The HOSC is recommending iPerf3 for Ku-IPS TCP and UDP testing for Windows and Linux.
 - TReK will host iPerf3 on their Windows and Linux on-board Demonstration Payloads.
 - The HOSC will host iPerf3 in our schedulable Ku-IPS IVV Lab on Win7 and Linux systems.
- PDs are encouraged to consider iPerf3 in support of controlled, repeatable, and shareable test results.

Fragmentation (1)

- Prior to Ku-IPS payloads didn't face a fragmentation concern.
- Under Ku-IPS, if fragmentation is not allowed then packets that are near the maximum transmission unit (MTU) limit are at risk of being discarded.
 - The discard can occur at the HOSC or the remote site when an IPSec header is prepended to the packet in support of secure delivery.
- Failure to allow fragmentation has created situations where Payload Developer (PD) commands that had been working began to spontaneously fail with no identified change made by any party.
 - The root cause stems from PDs typically having no control over how the OS or COTS inserts data into IP packets.

Fragmentation (2)

- Updates to the *Payload to Generic User Interface Definition Document* (SSP 50305) and the *Payload Rack to ISS Software ICD* (SSP 52050) are underway to incorporate recommendations that fragmentation be allowed on-board and on the ground.
- The HOSC does not commit to support any further troubleshooting for payloads where fragmentation is not allowed on-board and on the ground.

Fragmentation (3)

- If desired, a PD can enable fragmentation at the command line to allow time for confidence to be gained before fragmentation is made the boot time default.
- **Be Advised:** If fragmentation is enabled at the command line then it only applies to new connections from that point forward.
- The method to enable fragmentation is OS specific.

MTU Reduction as an Alternative to Fragmentation

- As an alternative to fragmentation, it is possible to reduce the default Ethernet MTU from 1500 bytes to something smaller such as 1400 bytes.
- However, given that IP headers or trailers can be added anywhere along the data path the HOSC considers MTU reduction a brittle solution that creates a risk of data loss.
 - Since additional headers or trailers can be added along the delivery path the determination of a MTU that would always work is problematic.
- The HOSC does not recommend MTU reduction as a permanent solution.
 - It has and can be used on a short-term basis until fragmentation is enabled.

Path MTU Discovery as an Alternative to Fragmentation

- Path MTU Discovery is a possible alternative to fragmentation.
- However, in practice the required ICMP packets that Path MTU Discovery requires are often administratively blocked.
- The HOSC does not recommend Path MTU Discovery.

Ku-IPS Round Trip Time (RTT)

- The HOSC Ku-IPS to ISS RTT is 650 ms +/- ~50 ms.
- The PD can determine their RTT to HOSC by pinging the HOSC VPN GW from their remote site. This RTT must be added to the HOSC-ISS RTT to determine the Total RTT.
 - The PD's Total RTT will be on the order of 900 ms.
 - **A Total RTT of ~900 ms is likely well outside the PD experience base and has major implications for TCP data flows. See also BDP and BER for details.**

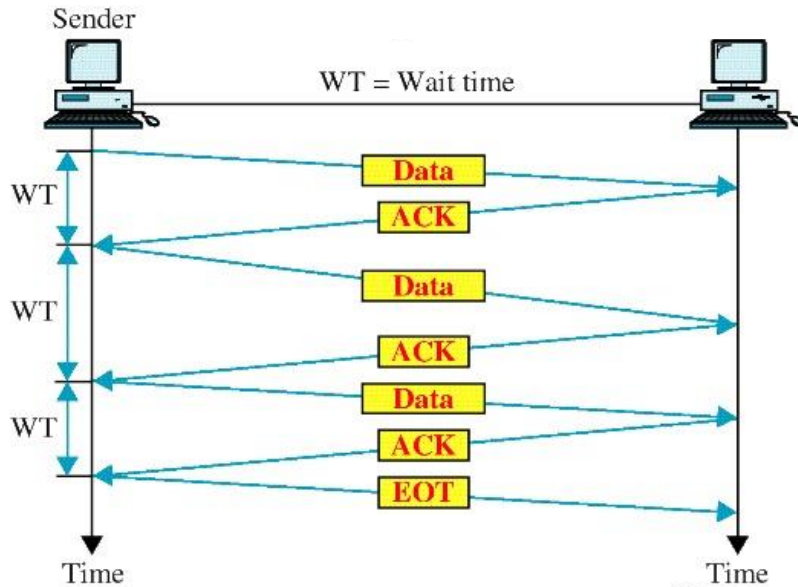
TCP Bandwidth Delay Product (BDP)

- Bandwidth Delay Product (BDP) calculations are used to estimate the required TCP Window Size necessary to achieve a given performance level. RTT and Bit Error Rate (BER) are major components of BDP.
- Failure to provide a TCP Window Size adequate to support the estimated BDP **will** result in reduced throughput for TCP data flows by making TCP operate as a stop and wait protocol.
- The default TCP Window Size is likely inadequate for Ku-IPS.
 - To achieve good TCP performance the characteristics of the specific data flow **must** be considered.
 - A 1st order approximation of Ku-IPS TCP Window Size is 1 MB.

Bit Error Rate (BER) & Packet Loss Rate

- Higher bit error rates drive the need for larger TCP buffers.
 - Robust BDP calculators include BER as a term.
- The HOSC does not have a good estimate of the Flight Ops Ku-IPS uncorrected BER. Currently we are assuming a 10^{-6} packet loss rate.
 - Ratty comm is typical near AOS/LOS transitions which means BER will typically be higher near AOS/LOS transitions.
- Currently the packet loss rate in the HOSC Ku-IPS IVV Lab is worse than the Flight Ops and ORT packet loss rate.
 - Tests with 1400 byte UDP packets have identified a HOSC Ku-IPS IVV Lab packet loss rate of 0.0% - 0.5% with higher data rates typically having a larger loss rate. The HOSC plans to investigate and resolve this anomaly.

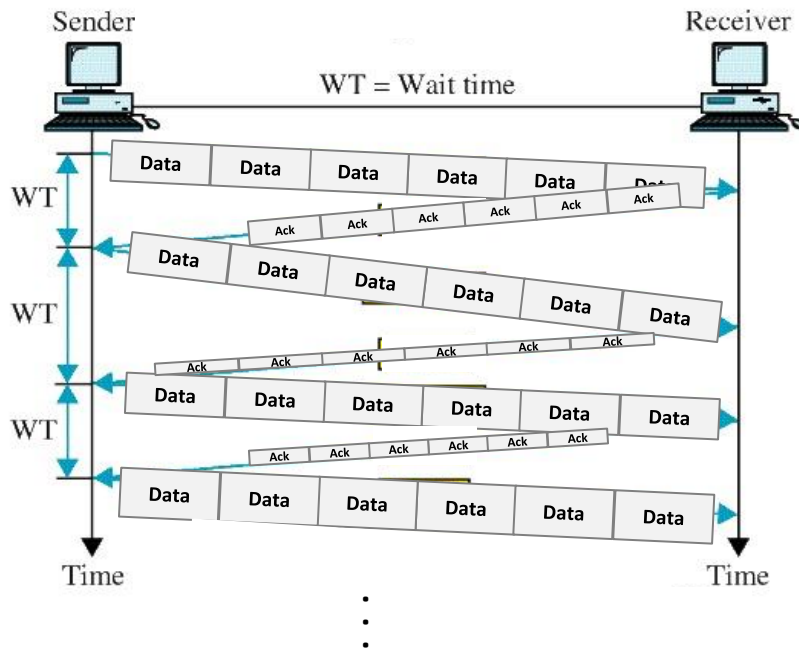
TCP Stop & Wait Example



The depicted TCP example would have very poor performance over the space link.

See the following slide for how to utilize TCP Window Size to avoid TCP Stop & Wait behavior.

TCP Streaming Example



The depicted TCP example would have good performance over the space link absent a large BER.

To avoid the throughput limitations created by a Stop & Wait protocol it is necessary to support sufficient data packets in transit such that the data path is kept full. To do so requires a large enough TCP Window Size to maintain state for all the packets in transit until a TCP acknowledgement for each can be returned from the receiver at the remote end to the sender.

Configuring TCP Window Size on Windows Systems

- For Windows 7 we have not yet found a way to change the default 64 KB TCP Window Size.
 - We are searching for viable approaches.
- iPerf3 `–w` command line option sets the TCP window size. This suggests a method could be reverse engineered from the iPerf3 source code though this approach is only viable when access to the source code is available.

Ku-IPS Baselined Protocol:

Remote Desktop Protocol (RDP)

- RDP characterization and tuning is likely to be a long-term collaborative project.
- RDP is likely to be a challenge over the space link and will perform poorly if BDP is inadequate.
- Microsoft provides an RDP server and client.
 - The HOSC plans to characterize RDP between a Win7 client and a Win7 server.
- There are multiple Linux RDP clients with each PD free to choose their own.

RDP Tuning Suggestions for Microsoft Windows (1)

- The Microsoft information that follows is nominally for Win7 clients connecting to a Win7 server and was assembled from Internet searches as well as various conversations and e-mails.
 - PD results and real world configuration experience is requested and appreciated for inclusion in the Ku-IPS Lessons Learned.

RDP Tuning Suggestions for Microsoft Windows (2)

- Set the RDP client display dimensions and color depth to the smallest values you can tolerate.
- Set Experience Performance to Satellite and then tweak settings as indicated herein.
- Enable persistent bitmap caching.
- Enable desktop composition (if available).
- Enable visual styles if (available).

RDP Tuning Suggestions for Microsoft Windows (3)

- Don't share local devices and resources you don't absolutely have to.
- Disable wallpaper.
- Disable remote audio and remote audio recording.
- Disable desktop background.
- Disable font smoothing.
- Disable the showing of window contents while dragging window.
- Disable menu and window animation.

RDP Tuning Suggestions for Linux Clients

- The HOSC has no current plans to test Linux RDP implementations and would appreciate tuning recommendations from PDs for inclusion in the Ku-IPS Lessons Learned.

Ku-IPS Uplink Data Rate

- Uplink Rate - A maximum of 8 megabits/sec. is allocated to the HOSC. The HOSC manages and schedules this bandwidth for all payloads.
- Uplink Shaping – NASA performs no traffic shaping on the uplink. If the aggregate payload uplink traffic exceeds 8 megabits/sec. the overage is clipped and dropped.
 - This implies PDs must support a traffic control mechanism.

Ku-IPS Downlink Data Rate & Traffic Shaping

- Downlink Rate:
 - Rates above the Payload Integration Agreement (PIA) are available on a scheduled basis.
 - Higher data rate users typically use APIDs or UDP protocols such as CFDP and Bundle Protocol (BP) for downlink.
- Downlink Shaping:
 - PDs are expected to stay within their PIA scheduled allocation which implies a rate control mechanism.

Ku-IPS Baselined Protocol:

Ping

- Ping makes for a good and simple initial test case.
- Pings to an on-board device provides a direct measurement of the Total RTT.
- Ping with a large buffer size (e.g., 60,000 bytes) demonstrate IP fragment handling.

UDP

- UDP can be used in support of basic throughput and loss tests without the complexities of TCP.
- iPerf3 can be used to support UDP tests.

Ku-IPS Baselined Protocol:

CFDP

- 2-node and 4-node CFDP was successfully tested bi-directionally during an ORT with the JSL @ SDIL.
 - 2-node native CFDP involved a payload node and a ground station node.
 - 2-node CFDP over BP testing involved a payload node and a ground station node.
 - 4-node testing CFDP over BP added an on-board gateway and a HOSC gateway.
- Forward work is planned to develop recommended native CFDP settings for ACKs, NACKs, and timeouts.

Ku-IPS Baselined Protocol: DTN (LTP & BP)

- Tests are planned to characterize the Licklider Transmission Protocol (LTP).
- Tests are planned to characterize Bundle Protocol (BP) BPING.

Ku-IPS Baselined Protocol: HTTPS

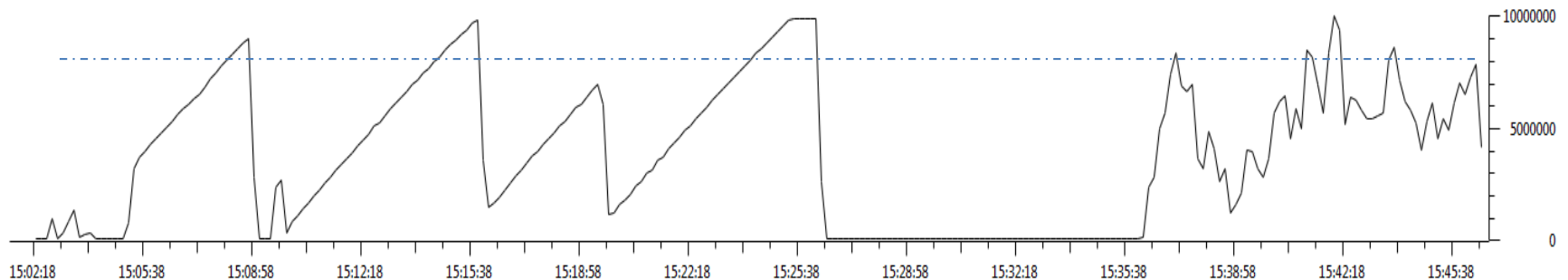
- The HOSC has done little HTTPS testing.
- The HOSC knows of one upload case where HTTPS performance was severely limited by inadequate TCP Window Size.

QNAP NAS

- What is conceptually simple may not be operationally simple.
 - Focused tests are recommended to baseline the NAS configuration.
 - The QNAP NAS wasn't developed with the space link in mind.
- Initial NAS tests yielded very limited throughput results.
- A recent retest involved uploading a single image file to a QNAP NAS in the HOSC Ku-IPS Lab to evaluate performance implications of a larger TCP Window Size.
 - While this test demonstrated a throughput of ~3 mb/s it also provided an unexpected result. Additional tests are planned but not scheduled.
 - Instead of the expected single 443/TCP session to upload the image there were 215 separate 443/TCP connections using the experimental SPDY protocol. SPDY may not be a good choice for the space link.
- NAS downlink tests are planned but not scheduled.

QNAP NAS Stop & Wait Example

- The HOSC NAS is the same make and model as the on-board NAS and considered to be in the same configuration.
- The plot below represents the aggregate throughput of 215 TCP connections, not the expected single connection!
 - While ~95% of the TCP connections were stalled we assume they tied up memory on both ends of the connection.
- The scale is bits/second before rate throttling. The dashed line is at the maximum uplink rate of 8 megabits/sec.
- Ideal performance would be a square wave at 8 megabits/sec.
- The ~10 minutes of near zero throughput is curious.



Backup Charts